## CHAPTER TWO: LITERATURE REVIEW

### Review of System 1: Stock Price Prediction Using Deep Learning (LSTM)

1. DESCRIPTION OF SYSTEM

- Overview of the system.

  The LSTM-based stock price prediction system utilizes deep learning techniques to predict stock prices. LSTM (Long Short-Term Memory) is a type of recurrent neural network (RNN) designed for time-series forecasting. The system learns from historical stock prices and other relevant data to predict future prices. LSTM is particularly well-suited for sequential data, which is essential for modeling stock price movements over time (Hochreiter & Schmidhuber, 1997).

- Models of the system
  - Context Model: The system operates in the context of stock market forecasting, taking historical stock prices and external economic indicators as inputs. The system outputs predicted stock prices for future time steps (Bao et al., 2017).
  - Interaction Model (Sequence Diagram): The sequence diagram shows the flow of data between the system and its components, including data preprocessing, model training, prediction generation, and output delivery to the user interface (Selvin et al., 2017).
  - Structural Model: The system consists of several layers: a data collection module, a data preprocessing module, the LSTM model, and the prediction and visualization interface (Fischer & Krauss, 2018).
  - Behavioral Model: The system continuously collects new stock data, trains the LSTM model, and updates predictions. When a user inputs a stock symbol, the model is invoked to generate future price forecasts (Bao et al., 2017).
- Features of the system
  - Real-time stock data collection and preprocessing
  - LSTM model for predicting future stock prices
  - Visual presentation of stock price trends
  - Predictive accuracy measurement and model evaluation

- Development tools and development environment of the system
    - Programming Languages: Python
    - Libraries: Keras, TensorFlow, Pandas, Matplotlib
    - Development Environment: Jupyter Notebook, Anaconda

## **2**. REVIEW OF THE GOOD FEATURES

- High Accuracy: LSTM networks are highly effective for sequential data and can capture long-term dependencies in stock market data (Hochreiter & Schmidhuber, 1997).
- Flexibility: The model can be adjusted to forecast prices for different time periods (e.g., hourly, daily, or weekly) (Fischer & Krauss, 2018).
- Adaptability: The LSTM model can be fine-tuned with additional market data such as financial indicators, sentiment analysis, and news trends (Bao et al., 2017).

## 3. REVIEW OF THE BAD FEATURES

- Computationally Expensive: Training LSTM models requires significant computational power and time, especially when using large datasets (Fischer & Krauss, 2018).
- Data Sensitivity: LSTM models are sensitive to data noise and require high-quality, cleaned data for accurate predictions (Selvin et al., 2017).
- Overfitting Risk: The model can overfit to historical data, especially if the training dataset is not diverse enough (Bao et al., 2017).

## 4. SUMMARY OF THE SYSTEM REVIEW

The LSTM-based stock price prediction system is a powerful tool for financial forecasting due to its ability to model complex, sequential patterns. However, it is computationally expensive and sensitive to noisy data, which can affect prediction accuracy. Despite these limitations, the system’s adaptability and high accuracy make it an attractive option for stock price prediction (Hochreiter & Schmidhuber, 1997; Fischer & Krauss, 2018).

## **Review of System 2: Stock Price Forecasting Using ARIMA**

## 1. DESCRIPTION OF SYSTEM

- Overview of the system.

  The ARIMA (Autoregressive Integrated Moving Average) model is a statistical approach used for time-series forecasting. It is often employed for stock price prediction by leveraging historical data. ARIMA works by analyzing the past values of stock prices and identifying trends, seasonal patterns, and irregularities to predict future prices (Box et al., 2015).

- Models of the system
  - Context Model: The system uses historical stock prices to model future price movements. The context involves time-series forecasting, which predicts future stock prices based on observed data (Ariyo et al., 2014).
  - Interaction Model (Sequence Diagram): The sequence diagram shows the flow of historical stock price data being fed into the ARIMA model, where it undergoes preprocessing and is used to generate a forecast (Box et al., 2015).
  - Structural Model: The system has modules for data collection, preprocessing, ARIMA model training, and prediction generation (Asteriou & Hall, 2015).
  - Behavioral Model: The system identifies patterns in past data, uses them for forecasting, and updates the model with the latest available data to improve prediction accuracy (Ariyo et al., 2014).
- Features of the system
  - Simple to implement and understand
  - Effective for stationary data with linear trends
  - Incorporates past price data for forecasting future prices
- Development tools and development environment of the system
  - Programming Languages: Python
  - Libraries: Statsmodels, Pandas, Numpy
  - Development Environment: Jupyter Notebook, Anaconda

## 2. REVIEW OF THE GOOD FEATURES

- Simplicity: ARIMA is relatively easy to implement and does not require a large amount of data compared to deep learning models (Box et al., 2015).

- Effectiveness with Stationary Data: ARIMA works well with time-series data that is stationary, making it a good choice for stable markets with clear trends (Asteriou & Hall, 2015).

3. REVIEW OF THE BAD FEATURES

- Poor Performance with Volatile Markets: ARIMA struggles to predict in highly volatile markets, as it assumes stationarity in the data, which is often not the case in stock markets (Ariyo et al., 2014).
- Limited to Linear Trends: ARIMA is less effective when dealing with non-linear patterns or sudden market changes (Box et al., 2015).
- Data Preprocessing Requirement: ARIMA models require thorough preprocessing and transformation of data to ensure it is stationary (Asteriou & Hall, 2015).

4. SUMMARY OF THE SYSTEM REVIEW

The ARIMA model is a well-established method for stock price forecasting, especially for data with clear trends. However, it has limitations in handling volatile and non-linear data, making it less suitable for unpredictable stock market behavior. Despite these drawbacks, ARIMA remains a useful tool for specific types of time-series data (Box et al., 2015; Ariyo et al., 2014).

**Review of System 3: Facebook Prophet for Time-Series Forecasting**

1. DESCRIPTION OF SYSTEM

- Overview of the system.

  Facebook Prophet is an open-source forecasting tool developed by Facebook to handle time-series data. It is particularly useful for forecasting data that exhibits seasonal trends and irregular patterns, such as stock prices. Prophet works by decomposing time-series data into three components: trend, seasonality, and holidays/events. It is known for its ease of use, robustness to outliers, and ability to handle missing data (Taylor & Letham, 2018).

- Models of the system

    - Context Model: The system takes historical stock prices and other external factors (e.g., holidays or special events) as input to forecast future prices (Taylor & Letham, 2018).
    - Interaction Model (Sequence Diagram): The sequence diagram illustrates how historical stock price data and events are input into Prophet, which generates predictions (Sean et al., 2016).
    - Structural Model: The system includes modules for data preprocessing, model training, and prediction visualization (Taylor & Letham, 2018).
    - Behavioral Model: Prophet decomposes the time-series data into components, trains the model, and updates forecasts based on new data input (Sean et al., 2016).
- Features of the system
    - Robust to missing data and outliers
    - Decomposes data into trend, seasonality, and holiday effects
    - Provides uncertainty intervals for predictions
    - Easy-to-use interface for non-technical users
- Development tools and development environment of the system
    - Programming Languages: Python, R
    - Libraries: Prophet, Pandas, Matplotlib
    - Development Environment: Jupyter Notebook, Anaconda

## 2. REVIEW OF THE GOOD FEATURES

- Ease of Use: Prophet is designed to be user-friendly, even for non-experts, and requires minimal tuning (Taylor & Letham, 2018).
- Robustness: It handles missing data and outliers well, making it suitable for real-world financial data (Sean et al., 2016).
- Accuracy: Prophet is known for its accuracy in forecasting stock prices, especially when the data exhibits seasonal behavior (Taylor & Letham, 2018).

## 3. REVIEW OF THE BAD FEATURES

- Limited for High-Frequency Data: Prophet performs better with daily or monthly data rather than high-frequency minute-level stock prices (Taylor & Letham, 2018).

- Assumption of Similar Trends: The model assumes that future trends will resemble historical patterns, which may not always hold in rapidly changing markets (Sean et al., 2016).

## 4. SUMMARY OF THE SYSTEM REVIEW

Facebook Prophet offers a robust and easy-to-use solution for time-series forecasting, particularly in cases where data exhibits clear seasonal patterns. However, its reliance on historical trends for future predictions and its limitations with high-frequency data could hinder its application in certain stock prediction scenarios. Despite these drawbacks, Prophet remains a powerful tool for many forecasting tasks (Taylor & Letham, 2018; Sean et al., 2016).